# Table of Contents

Lab Overview - HOL-1921-02-CMP - vRealize Automation 7 - Advanced Topics ............2

Lab Guidance .......................................................................................................................3

Module 1 - Blueprinting Applications with vRealize Automation (60 minutes) ............9
  Introduction..........................................................................................................................10
  Creating Your Application Topology ..................................................................................11
  Using Cloud Client to Manage Content ...........................................................................26
  Working with Software Components ................................................................................32
  Parameterized Blueprints and OVF deployment ...............................................................60
  Conclusion..........................................................................................................................81

Module 2 - Blueprinting and Managing Network Topologies with vRealize Automation (60 minutes) ........................................................................................................83
  Introduction..........................................................................................................................84
  Getting Prepared ...............................................................................................................85
  Working with NSX Constructs in vRealize Automation .....................................................90
  Day Two NSX Operations .................................................................................................109
  Conclusion..........................................................................................................................121

Module 3 - Using vRealize Automation to Discover, Deploy and Manage Containers (60 minutes) ..........................................................................................................123
  Introduction..........................................................................................................................124
  Building Your First Container Based Application ...........................................................125
  Sharing Your Application .................................................................................................148
  Conclusion..........................................................................................................................159

Module 4 - Working with the vRealize Automation Health Broker (45 minutes) ............161
  Introduction..........................................................................................................................162
  Review the System Test .....................................................................................................163
  Review the Tenant Test .......................................................................................................170
  Validating Tenant Health ...................................................................................................175
  Health Broker Extensibility ...............................................................................................180
  Conclusion..........................................................................................................................192
Lab Overview - HOL-1921-02-CMP - vRealize Automation 7 - Advanced Topics
Lab Guidance

Note: It may take more than 90 minutes to complete this lab. You should expect to only finish 2-3 of the modules during your time. The modules are independent of each other so you can start at the beginning of any module and proceed from there. You can use the Table of Contents to access any module of your choosing.

The Table of Contents can be accessed in the upper right-hand corner of the Lab Manual.

Learn all about the advanced capabilities of vRealize Automation. This deep-dive covers more complex service authoring capabilities, including new NSX integration capabilities, creating applications using Containers, vRealize Automation Health Monitoring, creating and using Custom Forms, and much more.

Lab Module List:

- **Module 1 - Blueprinting Applications with vRealize Automation** (60 minutes) (Intermediate) Discover what lies beyond IaaS, and how you can use blueprint parameterization and Software Components to deliver real world benefits over and above simple virtual machine provisioning. Learn how to handle dependencies and pass information between components in multi-tier applications, and parameterize more complex scripts to allow for reuse across blueprints.

- **Module 2 - Blueprinting and Managing Network Topologies with vRealize Automation** (60 minutes) (Intermediate) Incorporate NSX network constructs to deliver end to end automation of your applications. From simple load balancing, to more complex topologies - consuming existing networks or deploying them at runtime. You will also explore some of the Day 2 capabilities to manage your NSX constructs.

- **Module 3 - Using vRealize Automation to Discover, Deploy and Manage Containers** (60 minutes) (Intermediate) Whether you want to provide container hosts in controlled environments, or discover and manage containers that already exist within your environment, we've got you covered. This module will give you everything that you need to get started with Container Management through vRealize Automation.

- **Module 4 - Working with the vRealize Automation Health Broker** (45 minutes) (Intermediate) Monitoring your environment is a key part of system administration. This module will show you how to take advantage of the new Health Broker service to monitor key parts of your vRealize Automation environment, and highlight any issues that would result in degradation of service. You'll even get to build and run your own custom monitoring solutions!

Lab Captains:
This lab manual can be downloaded from the Hands-on Labs Document site found here:

http://docs.hol.vmware.com

This lab may be available in other languages. To set your language preference and have a localized manual deployed with your lab, you may utilize this document to help guide you through the process:


**Location of the Main Console**

1. The area in the RED box contains the Main Console. The Lab Manual is on the tab to the Right of the Main Console.
2. A particular lab may have additional consoles found on separate tabs in the upper left. You will be directed to open another specific console if needed.
3. Your lab starts with 90 minutes on the timer. The lab can not be saved. All your work must be done during the lab session. But you can click the **EXTEND** to increase your time. If you are at a VMware event, you can extend your lab time twice, for up to 30 minutes. Each click gives you an additional 15 minutes.
Outside of VMware events, you can extend your lab time up to 9 hours and 30 minutes. Each click gives you an additional hour.

**Alternate Methods of Keyboard Data Entry**

During this module, you will input text into the Main Console. Besides directly typing it in, there are two very helpful methods of entering data which make it easier to enter complex data.

**Click and Drag Lab Manual Content Into Console Active Window**

You can also click and drag text and Command Line Interface (CLI) commands directly from the Lab Manual into the active window in the Main Console.

**Accessing the Online International Keyboard**

You can also use the Online International Keyboard found in the Main Console.

1. Click on the Keyboard Icon found on the Windows Quick Launch Task Bar.
Click once in active console window

In this example, you will use the Online Keyboard to enter the "@" sign used in email addresses. The "@" sign is Shift-2 on US keyboard layouts.

1. Click once in the active console window.
2. Click on the **Shift** key.

Click on the @ key

1. Click on the "@ key".

Notice the @ sign entered in the active console window.
Activation Prompt or Watermark

When you first start your lab, you may notice a watermark on the desktop indicating that Windows is not activated.

One of the major benefits of virtualization is that virtual machines can be moved and run on any platform. The Hands-on Labs utilizes this benefit and we are able to run the labs out of multiple datacenters. However, these datacenters may not have identical processors, which triggers a Microsoft activation check through the Internet.

Rest assured, VMware and the Hands-on Labs are in full compliance with Microsoft licensing requirements. The lab that you are using is a self-contained pod and does not have full access to the Internet, which is required for Windows to verify the activation. Without full access to the Internet, this automated process fails and you see this watermark.

This cosmetic issue has no effect on your lab.

Look at the lower right portion of the screen
Please check to see that your lab is finished all the startup routines and is ready for you to start. If you see anything other than "Ready", please wait a few minutes. If after 5 minutes you lab has not changed to "Ready", please ask for assistance.
Module 1 - Blueprinting Applications with vRealize Automation (60 minutes)
Introduction

This Module contains the following lessons:

• Creating Your Application Topology
• Using Cloud Client to Manage Content
• Working with Software Components
• Parameterized Blueprints and OVF deployment
Creating Your Application Topology

Before you can start getting fancy with deploying an application, you need to represent the topology. In this first lesson you will create a simple Web/Database topology that will be used as a basis to deploy Wordpress.

Remember that in order to create a new Blueprint from scratch, your account needs to have the Infrastructure Architect role assigned.

Open Chrome Browser from Windows Quick Launch Task Bar

1. Click on the Chrome Icon on the Windows Quick Launch Task Bar

Login to vRealize Automation

1. Enter cloudadmin as the username and VMware1! as the password.
2. Click Sign in.
Create a New Blueprint

1. Click on the **Design** tab.
2. Select **Blueprints**.
3. Click the **New** button.
Enter Blueprint Details

Under the General tab, fill out the Blueprint details as follows:

1. In the Name field, enter Two Tier Wordpress. Note that the ID will fill out automatically for you.
2. Enter a useful Description. This field is queried when you do a search from the catalog, so it should contain keywords that a user may search with.
3. Set the Lease (days) values to a Minimum of 1 and Maximum of 30.
4. Click on NSX Settings.
Define Transport Zone

1. Click on the Transport zone and select RegionA0_Global_TZ (vCenter).
2. Click on OK.

Add Basic Networking

2. Drag and Existing Network onto the design canvas.
3. When the configuration pane pops up, click the Network profile ellipses.

The first step when creating a topology is to bring across the required network components. While having these objects on the canvas is not a hard requirement to consume portgroups out of vCenter, it is a good habit to get into. As a topology gets more complex, being able to see the relationships between networks and VMs is helpful for visualization and troubleshooting.

2. Drag and Existing Network onto the design canvas.
3. When the configuration pane pops up, click the Network profile ellipses.
Assign a Network Profile

It is worth pointing out that you are selecting a **Network Profile** from this screen, not a **Network Path** (typically a port group).

Your **Network Profiles** are bound to a given **Network Path** in your Reservation. The takeaway here is that the name and description of your **Network Profiles** should make it easy for blueprint designers to understand. In some organizations that may be a VLAN ID, in others it may need to be more descriptive - such as "Public Web DMZ".

1. Select **Default External Network Profile**.
2. Click **OK**.
3. Click on the design canvas to minimize the dialog, and then click **Save**. (Not Shown)

Do not click **Finish** when you have finished configuring a Blueprint component. This will save your Blueprint and take you back to the Blueprints page!
Add the Web Server Component

Next up you will add a Virtual Machine for the Web Tier.

1. Under **Categories** select **Machine Types**.
2. Scroll to the bottom and drag a **vSphere (vCenter) Machine** component onto the design canvas.

The configuration properties will immediately pop up. Resize the panel as needed to make it easier to work with.
Define the Web Server General Properties

Under the General tab, there are a few mandatory details that need to be set, and some that are optional.

1. While the **ID** value is set to a default value, it is strongly recommended that you set this to something meaningful. Not only does it make it easier to understand the composition of your Blueprint, it also impacts the labels assigned to components in the Execution Plan at request time. This results in a better user experience and much greater ease of troubleshooting. Set this value to **Web_Server**.
2. This environment is making use of **Reservation Policies** to inform the Reservation selection process. From the **Reservation Policy** dropdown, choose **VM Reservation Policy**.
3. Change the **Maximum** instance value to 2.
4. Finally, click the **Build Information** tab.
Define the Web Server Build Properties

1. Set the **Action** to **Linked clone**.
2. Ensure that **CloneWorkflow** is selected in the **Provisioning workflow** dropdown.
3. Click on the ellipses at the end of the **Clone from** field and select **base-linux-cli** from the resulting dialog (not shown).
4. Select **application authoring snapshot** from the **Clone from snapshot** dropdown.
5. In the **Customization spec** field, type **CentOS**. (Note: This field is case sensitive)
6. Once you’ve finished with the **Build Information** tab, select the **Network** tab.

The **Build Information** tab is where you define the means by which a workload is provisioned. The **Blueprint Type** is simply a means of tracking license consumption, and is not used in this lab.
Define the Web Server Network Properties

The **Network** tab is where you have the ability to apply configuration or constraints to the network configuration of your Blueprint component. As a simple example, you may want to prevent users from adding additional network interfaces when they select the **Reconfigure** Day 2 Action. In that case, you could set the **Maximum network adapters** value to be 1 (or your preference).

You also have the option here to determine whether the network adapter will use DHCP, or assign a static IP address from a network profile or IPAM system.

1. To add a NIC to configure, click on the **New** button.
2. From the **Network** dropdown, select **DefaultExternalNetworkProfile**. Note that if you don't have any networks on the canvas, this dropdown will be blank.
3. Click on **OK**.

You'll note that the moment you click **OK**, a line connecting the virtual machine component to the network component appears.

4. Click on the design canvas to minimise the dialog, and then click **Save**. (Not shown)
Add the Database Server Component

Next up you will add a Virtual Machine for the Database Tier.

1. Under Categories select Machine Types.
2. Scroll to the bottom and drag a vSphere (vCenter) Machine component onto the design canvas.

The configuration properties will immediately pop up. Resize the panel as needed to make it easier to work with.
Define the Database Server General Properties

Under the General tab, there are a few mandatory details that need to be set, and some that are optional.

1. Click on the General tab
2. Set the ID value to Database_Server.
3. From the Reservation Policy dropdown, choose VM Reservation Policy.
4. Finally, click the Build Information tab.
Define the Database Server Build Properties

1. Set the **Action** to **Linked clone**.
2. Ensure that **CloneWorkflow** is selected in the **Provisioning workflow** dropdown.
3. Click on the ellipses at the end of the **Clone from** field and select **base-linux-cli** from the resulting dialog. (Not shown)
4. Select **application authoring snapshot** from the **Clone from snapshot** dropdown.
5. In the **Customization spec** field, type **CentOS**.
6. Once you've finished with the **Build Information** tab, select the **Network** tab.
Define the Database Server Network Properties

1. To add a NIC to configure, click on the **New** button.
2. From the **Network** dropdown, select **DefaultExternalNetworkProfile**. Note: If you don't have any networks on the canvas, this dropdown will be blank.
3. Click on **OK**.

You'll note that the moment you click **OK**, a line connecting the virtual machine component to the network component appears.

4. Click on the design canvas to minimize the dialog, and then click **Save**. (Not shown)
Review Your Topology

Have a quick look at your topology to make sure that it’s what you want to build. Depending on where you placed your `Web_Server` and `Database_Server` components your design canvas may look a little different.

1. Click **Finish** to save and exit the blueprint canvas.

In terms of making your blueprinting process more scalable, you could make a number of topology centric blueprints, and then have them as the foundation for application blueprints in your own organization.
Publish Your Blueprint

Back on the Blueprints screen, your Two Tier Wordpress Blueprint should be at the top of the list.

1. Select it, and then click **Publish**.

With that, you are ready to move on to the wonderful world of Software Components. Don't logout - you'll need this session in the next lesson.
Using Cloud Client to Manage Content

While building an application blueprint is not an overly complex task, sometimes the time requirement to test component interoperability can be overwhelming. vRealize Automation provides the mechanism to share blueprints and their components, saving you the trouble of building all of your blueprints from scratch. In this lesson we will import some software components to speed up the process of creating your Wordpress blueprint.

Check for Existing Software Components

First of all, take a look at the Software Components that are already available to you.

1. Click on the **Software Components** tab to view them.

You'll note that there is an Apache Service, but not much else that can be reused. Wouldn't it be great if you could just import all of the Software Components required to build out the Wordpress Blueprint?
In a stroke of luck, a colleague of yours has shared some content with you from their own vRealize Automation environment. They conveniently placed it on your machine, in C:\hol\software.zip.

1. Open **File Explorer** from the Task Bar (not shown)
2. Navigate to **C:\hol\HOL-1921-02-CMP**
3. Double-click on **software.zip** to open the file

### Review Shared Content

1. Double-click on **metadata.yaml** and open it with Notepad from the resulting dialog. (Not shown)
2. Browse through this file. It lists the components included within the zip file, and also highlights any dependencies that they may have (as identified at the time of export). These dependencies impact the order in which components would be imported. Keep this dependency in mind, a little later on you will see how it manifests in the UI.

Close both the Notepad and File Explorer windows, and get ready to import this content!

**Launch CloudClient**

1. Minimize the web browser, and launch CloudClient from the desktop. It will take a moment to load completely.

It is important to note that when using CloudClient, roles are honored in the same way as they would be in the UI. This means that if you want to import content, you need to have the right role. This will differ for XaaS Blueprints (XaaS Architect), IaaS Blueprints (Infrastructure Architect) and Software Components (Software Architect/Infrastructure Architect).

**Login to vRealize Automation**

One of the big challenges with working on command line based tools is using them regularly enough to remember the command sequences! Thankfully CloudClient supports tab based autocomplete.
1. Type in `vra login userpass` -- and then press tab. You'll note that a full list of arguments is provided.

2. Complete the command to be as shown in the following code block and then press Enter. Remember that you can also highlight the text below and drag-and-drop it to the command window, rather than typing it.

   ```
   vra login userpass --server vra-01a.corp.local --user cloudadmin@corp.local --password VMware1! --tenant vsphere.local
   ```

### Import Blueprint Components

When performing an import, you have a few options. How do you want to handle conflict resolution - skip or overwrite? Would you like to perform a dry run and not actually import anything, just confirm that the import will work ok?

If you'd like to take a look at the available arguments, then you can use the tab autocompletion again. In this case, let's keep it simple:

1. Enter the following command:

   ```
   vra content import --path c:\\hol\\HOL-1921-02-CMP\\software.zip --resolution SKIP
   ```

   You will see the a successful response under Import Result if it has imported successfully.

---

HOL-1921-02-CMP
2. Close the CloudClient by typing `exit` and hitting `Enter`, and head back to the vRealize Automation web UI.

**Check for Imported Components**

Head back to your vRealize Automation console.

1. To make the imported content show in the inventory, click the **Refresh** button. You should now see the list of components shown above.
2. Click on the **Wordpress Services** Software Component to review it in more detail.
Review Wordpress Services

You will note that this Software Component has a **Container** value assigned - Apache Services. Can you remember back to the yaml file, where Wordpress Services has a dependency on Apache Services? This is the reason for that. It controls the placement options, and would prevent the dropping of this Software Component onto an unsupported web server such as IIS.

1. Click on **Cancel** to close this screen. You will have a chance to work through the different screens and configuration details of software components shortly.
2. Click **Yes** on the resulting dialogue box (Not shown)
Working with Software Components

In this lesson, you will learn how Software Components are built and integrated into an Application Blueprint.

Open Your Blueprint

1. Select the **Blueprints** tab.
2. Click on the **Two Tier Wordpress** link to open it for editing.
Add Database Server Software Components

1. Select **Software Components** under **Categories**. When you drag each software component onto the canvas, a dialog will popup to allow you to configure it. Simply click on the canvas to make that disappear for now.
2. Drag the **MySQL** Software Component onto the **Database_Server**.
3. Drop the **Wordpress Database** inside the MySQL instance from step 2.
4. Next, drag the **CentOS Configuration** onto the **Database_Server**.
5. Hold your mouse over the **MySQL_1** component until you see a little circle appear. Drag that circle up the the **CentOS_Configuration** component. This creates a dependency, saying that **MySQL_1** cannot execute until **CentOS_Configuration** successfully completes.
Add Web Server Software Components

As with the previous step, each time you drag a software component onto the canvas, a configuration dialog will appear. Click on the canvas to make it disappear.

1. Drag the **PHP Service** onto the **Web_Server**.
2. Next, drag **Apache Services** (note the plural, do not drag Apache Service by mistake) onto the **Web_Server**.
3. Drag **Wordpress Services** inside the **Apache_Services_1** component.
4. The final Software Component to add in is **CentOS Configuration**, which should be dragged and placed directly onto the **Web_Server** component.
5. Create a dependency by dragging the **PHP_Service_1** circle up to **Apache_Services_1**.
6. Create another dependency by dragging the **Apache_Services_1** circle up to **CentOS_Configuration**.
Review Your Application

Before saving, take a look at your work and make sure that you have all the required components and dependencies mapped correctly. Note that your blueprint may look slightly different, depending on where you dropped the objects on the canvas.

If you're struggling to see the whole canvas, make use of the zoom buttons in the top right hand corner.

1. Click **Save** before you go any further.
Updating Wordpress Services

Because we imported a number of Software Components, and not a complete Blueprint, some dependency values need to be defined.

1. Select the **Wordpress Services** Software Component on the Web Server.
2. Select the **Properties** tab.
3. You'll note that the WP_DB_HOST property defaults to localhost, but we need it to point to our database server. Double click on **localhost** to edit it.

**Bind Database Host Value**
As mentioned previously, you need the WP_DB_HOST value to be the IP address of the database server. The problem is that you don't know this value ahead of time. Even if you did, you would not want to hardcode that value here as it would lead to any new deployments of the blueprint connecting to the same database server. Enter **Binding**.

1. Select the checkbox to make the property **Binding**.
2. Next, to identify the property that you want to bind to. As the IP address isn't determined until provisioning occurs, start with **_resource** and then hit ~. This will populate all available options.
3. Select **Database_Server**, then hit ~ again and choose **ip_address** (not shown).
4. Click **OK**.

With that, you have a property that will be identified at build time and thus will always be correct. Pretty neat, right?

Note that if you were to bind to a property that was available before provisioning, then you could skip straight to **Database_Server** without including **_resource**. The **_resource** prefix means that the property is pulled from the resource that is provisioned by the component in question, as opposed to coming from the component metadata itself.

5. Be sure to click **Finish** on the design canvas before continuing. (Not shown)

**Taking Stock**

At this point you have an application blueprint that you can publish to the catalog and provision. The result will be a working Wordpress environment that will present you with the configuration screen. In the next steps you will get a little more hands on with the creation of a new Software Component to handle the automatic configuration of the options you would be presented on the aforementioned configuration screen.
Creating a New Software Component

Let's begin creating your new Software Component.

1. Select the **Software Components** tab.
2. Click on the **New** button.
Defining General Properties

1. In the name field, enter **Wordpress Site Config**.
2. In the **Description** field, enter **Wordpress Site Configuration**.
3. To ensure that this component isn't mistakenly placed somewhere it shouldn't be, set the **Container** value to **Wordpress Services**.
4. Click **Next**.
Setting Up Script Properties

There are a few variables to define on this page that will be used in the script.

To create them:

1. Click **New**.
2. Enter the details as defined in the table below and click **OK**. Note that you can resize the Lab Manual pane in your browser to make this table more readable. Note that the "**admin_password**" property has no Type set. When you enter it, you can leave it set to **String**, but when you check the **Encrypted** box, it will automatically change to **Secure String**.
3. When you have entered all of the properties, click **Next**. (Not shown)

<table>
<thead>
<tr>
<th><strong>Name</strong></th>
<th><strong>Type</strong></th>
<th><strong>Value</strong></th>
<th><strong>Encrypted</strong></th>
<th><strong>Overridable</strong></th>
<th><strong>Required</strong></th>
<th><strong>Computed</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>site_ip</td>
<td>string</td>
<td></td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>wordpress_site_files</td>
<td>string</td>
<td>/var/www/html</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>admin_password</td>
<td>Secure String</td>
<td>****</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>admin_email</td>
<td>string</td>
<td><a href="mailto:cloudadmin@corp.local">cloudadmin@corp.local</a></td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>admin_user</td>
<td>string</td>
<td>admin</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>site_title</td>
<td>string</td>
<td></td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>
Defining Actions

The Actions screen is where you define the scripts that you want to run at specific stages of the workload lifecycle.

**Install** - initiates once provisioning and customization is complete. Note that **Configure** and **Start** are loosely couple with **Install**, and allow for multiple executions separated by a controlled reboot.

**Configure** - after a controlled reboot using the checkbox on the **Install** action, you can launch an additional set of commands. You can issue a second controlled reboot using the checkbox at the end of the action dialog.

**Start** - the final stage in which you can instantiate a set of commands during the initial provisioning phase. Will commence after a controlled reboot from the **Configure** action.

**Update** - executes during Scale In or Scale Out Day 2 Actions.

**Uninstall** - executes when a machine is manually destroyed.

1. Your script will run at the **Install** stage. Click as instructed to open the dialog box, and then paste in the snippet below using either copy and paste, or the built in drag-and-drop functionality.

```
#!/bin/bash
```
echo "# --- Display Environment Variables for Debugging"

env

echo ""

echo "# ---"

echo ""

php /tmp/wp-cli.phar --info

mv /tmp/wp-cli.phar /usr/local/bin/wp

wp core install --url=http://$site_ip --path=$wordpress_site_files --title="Welcome to the home of the $site_title" --admin_user=$admin_user --admin_password=$admin_password --admin_email=$admin_email

Using Properties

Before you close the screen, it's worth noticing that your syntax is highlighted to make it easier to read and understand.

1. To demonstrate another handy capability, select $site_ip and delete it.
2. Click the **Select a property to insert** dropdown and choose **site_ip**. This list is the set of properties you defined in the previous step.
3. Click **OK** to continue.

Click **Next** when you return to the main screen, and then **Finish** (not shown).
Publish Your Software Component

1. Ensure that your Software Component is selected.
2. Click Publish.
Open Your Wordpress Blueprint

To add the newly created Software Component, you need to re-open the Wordpress Blueprint.

1. Select **Blueprints** from the left menu.
2. Click on the **Two Tier Wordpress** hyperlink.

Allowing the Requestor to Specify a Site Name
One important thing to consider when it comes to user defined inputs is where they exist on the request screen. If you were to simply tick the **Show in Request** checkbox on the Software Component that you are about to add to your Blueprint, the user would have to click quite deep into the request to set the desired value. To make this a little more intuitive, you will create the Custom Property on the Web Server component.

Select the Web Server to bring up the configuration page.

1. Click on the **Properties** tab.
2. Select the **Custom Properties** tab.
3. Click **New**.
4. Enter **SiteTitle** as the Custom Property **Name**.
5. Tick the **Show in Request** check box,
6. Click **OK**.
7. Click on the Design Canvas to minimize the configuration panel (not shown)
8. Click **Save** (not shown).

### Add the Software Component

1. Select the **Software Components** category
2. Drag the **Wordpress Site Config** Software Component onto the **Wordpress_Services_1** component on the **Web_Server**.
Binding the Dynamic Properties

Just like before, there are a couple of properties that need to be bound to get their value.

1. Select the **Properties** tab
2. Select the **site_title** property, and click **Edit**.
3. Check the **Binding** checkbox
4. Type in **Web_Server~SiteTitle** - this binds the property to the Custom Property created a few steps back.
5. Click **OK** to save the property.
6. Repeat steps 2 through 5 for **site_ip**, binding to **_resource~Web_Server~ip_address**.
7. When you have completed that, click **Finish** (not shown).

Your Blueprint is ready to publish to the catalog!
Entitle Your Blueprint

1. Click on the Administration menu.
2. Select Catalog Management (Not shown), and then Catalog Items.
3. Locate your Two Tier Wordpress blueprint in the list and click on the hyperlink to open up the Catalog Item.
Add Catalog Item to Service

1. Using the **Service** drop down, assign the Catalog Item to the **Infrastructure** Service.
2. Click on **OK** to save your changes and exit.
   Finally, you can request your new shiny Wordpress Application!
3. Logout from your cloudadmin session using the **Logout** button in the top right corner, then click the **Go back to login page** button (Not shown)
Login as DevUser

1. Enter **devuser** as your username and **VMware1!** as your password.
2. Click **Sign in**.

Request Wordpress Blueprint
Set Site Name

1. Click on the Catalog tab.
2. Locate the Two Tier Wordpress Catalog Item and click Request.

1. Click on the Web_Server.
2. Set the SiteTitle value to Succulents
3. Click Submit.
4. Click OK on the confirmation (not shown).
Check Request Progress

1. To track the progress of your request, click on the Requests tab.
2. Select your request and click View Details.
To get detailed information about the progression of your request, click on the **Execution Information** button.

As Software Components succeed or fail, you can review their output by clicking on the ellipses.

Click **Refresh** to update the status of the components. (This could take 10-15 minutes to complete)

Click **OK** when all components have successfully provisioned.
Get Web Server IP

To confirm that everything has provisioned correctly, you'll want to check that you can access the Wordpress site.

1. Select the **Items** tab.
2. Click the triangle to expand your Deployment.
3. Identify the component that is the **Web_Server** and make note of the **IP Address** value.

Note that the IP Address may not be identical to the one shown in the image if you completed other modules prior to this one.
Validate Wordpress is Online

Open a new browser tab, and enter the IP Address you noted in the previous step.

You will see the webpage as shown above, the front end of your Wordpress application.

Scaling Out

1. Select the 'Two Tier' deployment.
2. Click on the 'Actions' dropdown and select 'Scale Out'.
So now you've managed to deploy a two tier application complete with software components.

Imagine for a moment that your Wordpress blog is an overnight success and you need more than just a single web host to serve all the hits you are getting.

Enter the Scale Out Day 2 Action. While you will trigger this manually now, it is possible to initiate this with the API, using an external trigger such as vRealize Operations.

1. Switch back to the vRealize Automation tab, then locate and select your Deployment. Be sure not to click on the blue text, or you will go into the Details view.
2. Click Actions and then select Scale Out.

### Add Scale Out Details

Since the Scale Out Action is a Deployment level Action, you can scale out multiple nodes in a single request if your Blueprint supports it. In this case, only the Web Server is configured to support more than one node.

1. Select the **Web Server**.
2. Change the **Scale out to** value to **2**.
3. Click on the **Submit** button.
On the confirmation dialog that pops up (not shown), click on the **OK** button. Note that this dialog details the specific changes that will be made to your deployment when executing the Scale operation.

Click **OK** again on the request submission dialog (not shown).

**Monitor Your Request**

1. To monitor the progress of your request, click on the **Request**s tab.
2. Next, locate and select your **Scale Out** request, clicking the **View Details** button to open it (not shown).
3. Select the **Execution Information** button to get the extra detail of the request.
4. You’ll notice that there are two different request types occurring - **Update** and **Provision**. Why is that? If you remember back to creating your own software components, there was an Action state called **Update**. If you were to have a script in that action, this is when it would be executing, on the existing Web Server. You also have the **Provisioning** of the new Web Server.
5. Once all of the Status values change to **Successful**, click on **OK**.
Check the New Web Server IP

Like you did previously, get the IP Address of the new Web Server.

1. Select the **Items** tab.
2. Click on the triangle to expand out the Deployment.
1. Open a new browser tab, and enter the IP Address you noted in the previous step.

You will see the webpage as shown above, the front end of your Wordpress application.

If you were going to scale out in the real world, you would want a load balancer in the mix. In the next module you will learn about how you can integrate network and security components into your application blueprint to deliver a holistic application stack.
Destroying Your Deployment

Now that you've successfully deployed your Wordpress application, it is time to tidy up and move onto the next lesson.

1. Click on the **Items** tab.
2. Select your Deployment, click on **Actions** and then click **Destroy**.

On the page that pops up, click the **Submit** button (not shown) to confirm your request. Once that completes, click on the **Logout** button in the top right corner of the browser (also not shown).
Parameterized Blueprints and OVF deployment

In this lesson you will learn how to use component profiles to create parameterized blueprints and how to deploy an OVF/OVA appliance using a blueprint.

Navigate to the Property Dictionary

First let's take a look at the Component Profiles which are used to parameterize blueprints. Component Profiles are located within the Property Dictionary. To navigate to the Property Dictionary:

1. Select the Administration tab
2. Click on Property Dictionary
View Component Profiles

1. Select **Component Profiles**

There are two different types of Component Profiles;

- **Image** - Allows different images to be selected during the deployment. e.g. you may want to use the same blueprint to deploy a machine image to either a development environment or a production environment. However each environment may require a different image. Using an Image Component Profile allows for a single blueprint to be used rather than having to create specific blueprints for each environment. This helps to reduce the number of blueprints that need to be managed.

- **Size** - Allows the administrator to have more control over the size of deployments. Rather than allowing minimum and maximum values for cpu, memory and storage, allowing to a user to select anything within these definitions. Using the Size component profile allows the administrator to set t-shirt sizes with predefined values for cpu, memory and storage.

2. Select the **Size** Component Profile
Add a Value Set

1. Click the **Value Sets** tab

You will see that we have already defined Value Sets named Medium and Large. Now let's create a new one called Small.

2. Click **New**
Define the new Value Set

1. Let's give the Value Set a name, this will be the name that the end user sees. Enter **Small** for the Display Name
2. Now specify the following values for our Small Value Set
   - CPUs: **1**
   - Memory (MB): **2048**
   - Storage (GB): **16**
3. Select Finish to save the changes and return to the previous page
Create a new OVF Based Blueprint

Now its time to create a Blueprint which deploys and OVF image.

1. Select the Design tab
2. Ensure that Blueprints is selected
3. Click +New
Enter Blueprint details

1. In the Name: enter HOLDemoOVF
2. Click OK

Add Basic Networking

[Diagrams and steps for adding a network to the Blueprint]

Let's first add a network to the Blueprint
2. Drag and Existing Network onto the design canvas.
3. When the configuration pane pops up, click the Network profile ellipses.

Assign a Network Profile

It is worth pointing out that you are selecting a Network Profile from this screen, not a Network Path (typically a port group).

Your Network Profiles are bound to a given Network Path in your Reservation. The takeaway here is that the name and description of your Network Profiles should make it easy for blueprint designers to understand. In some organizations that may be a VLAN ID, in others it may need to be more descriptive - such as "Public Web DMZ".

1. Select Default External Network Profile.
2. Click OK.
3. Click on the design canvas to minimize the dialog, and then click Save. (Not Shown)

Do not click Finish when you have finished configuring a Blueprint component. This will save your Blueprint and take you back to the Blueprints page!
Add a vSphere Machine Component

Next up you will add a Virtual Machine for the OVF Appliance

1. Under Categories select Machine Types.
2. Scroll to the bottom and drag a **vSphere (vCenter) Machine** component onto the design canvas.

The configuration properties will immediately pop up. Resize the panel as needed to make it easier to work with.
Define the Server General Properties

Under the General tab, there are a few mandatory details that need to be set, and some that are optional.

1. While the ID value is set to a default value, it is strongly recommended that you set this to something meaningful. Not only does it make it easier to understand the composition of your Blueprint, it also impacts the labels assigned to components in the Execution Plan at request time. This results in a better user experience and much greater ease of troubleshooting.
   Set this value to **HOL_Demo_App**.
2. Finally, click the **Build Information** tab.

Define the Build Information
Here we define the OVF information. The ImportOVFWorkflow supports both .ova and .ovf file formats. The files need to be hosted on a html server. In this use case we have hosted a demo ovf appliance on util-01a which is a utility vm within your Lab environment.

1. First select `ImportOVFWorkflow` from the Provisioning workflow drop down menu.
2. In the following url enter the following; `http://util-01a.corp.local/hol-demo-app-v0.1.ovf`
3. Click `Configure` which will allow you to accept the OVF EULA and configure the OVF properties

### Agree to the OVF EULA

![Configure OVF dialog]

1. Make sure you read the entire EULA, just kidding in this instance its fake. Click the checkbox to agree to the terms of the EULA
2. Click `Continue`
Configure the OVF Properties

At this point the OVF properties are imported in and you can update the property values.

1. Double click on appname to edit it and enter **AppX** as the value.
2. Click OK to save the setting

**Complete OVF Properties**
1. Now that you know how to edit the properties go ahead and add the following values;
   ◦ ip0 - **192.168.120.100**
   ◦ color - Enter one of the following; BLUE, GREEN, PINK this value is case sensitive
   ◦ mood - Enter one of the following; AWAKE, TIRED, CONFUSED, HUNGOVER this value is case sensitive

2. Click **OK**

**View the Properties**

The properties that we just imported from the ovf file are stored as custom properties. Let’s take a look;

1. Click the Properties tab
2. Click on Custom Properties
3. Click Profiles
Assign the Component Profile

Now let's assign the tshirt sizes we viewed and created earlier to the blueprint.

1. Click +Add, not shown in image
2. Place a tick Size check box
3. Click OK

Edit the value sets

1. Select the Size component profile by clicking on the row
2. Click Edit Value Sets
Select Value Sets

Here you can choose which value sets are available on blueprint request. You can also set the default option.

1. Go ahead and select all available value sets by selecting the top checkbox
2. Next let's set the default option to Medium
3. Click OK

Assign the Network Profile

Finally let's not forget to attach the network profile to our machine.

1. Click on the Network tab
2. Click New
3. Select DefaultExternalNetworkProfile from the dropdown list
4. Click OK
5. Finally click in on the design canvas

**Review the finished Blueprint**

Take a look at the blueprint on the canvas, it should look like this.

1. Click Finish

**Publish the Blueprint**

1. Select the blueprint HOL_Demo_App by clicking on the row, do not click on the hyperlink as it will open the blueprint
2. Click Publish

**Entitle Your Blueprint**

![Image of vRealize Automation interface]

1. Click on the Administration menu.
2. Select Catalog Management (Not shown), and then Catalog Items.
3. Locate your HOL_Demo_App blueprint in the list and click on the hyperlink to open up the Catalog Item.

**Add Catalog Item to Service**

![Image of Catalog Item setting]

1. Using the Service drop down, assign the Catalog Item to the Infrastructure Service.
2. Click on OK to save your changes and exit. Finally, you can request your new shiny HOL_Demo_App.
3. Logout from your cloudadmin session using the Logout button in the top right corner, then click the Go back to login page button (Not shown)

**Login as DevUser**

1. Enter **devuser** as your username and **VMware1!** as your password.
2. Click **Sign in**.
Request the Blueprint

1. Click on the **Catalog** tab.
2. Locate the HOL_Demo_App Catalog Item and click **Request**.

Select the deployment size

1. Click on the **HOL_Demo_App** machine component.
2. Change Size to **Small** remember when you assigned the value sets you set the default as Medium
3. Click **Submit**.
4. Click **OK** on the confirmation (not shown).

### Check Request Progress

1. To track the progress of your request, click on the **Requests** tab.
2. Click the Refresh button until the status reads **Successful**, the deployment will take approx 5 minutes.
Validate the deployment

1. Open a new browser tab, and enter the IP Address **192.168.120.100** remember how this address was set for the OVF property IP value?

You will see a webpage as similar to the one above depending on what color and mood values you decided to set. This demonstrates how you were able to deploy an ovf appliance and set the ovf properties within the blueprint.
Destroying Your Deployment

Now that you've finished reviewing the OVF Deployment, it is time to tidy up. Head back to your vRealize Automation page.

1. Click on the **Items** tab.
2. Select your Deployment, click on **Actions** and then
3. Click **Destroy**

On the page that pops up, click the **Submit** button (not shown) to confirm your request. Once that completes, click on the **Logout** button in the top right corner of the browser (also not shown).
Conclusion

In this module you learned how to create an application topology and how to import and use software components to deploy a complete application. You also learned how to use custom properties in a request to set configurations within deployed applications and deploy ovf appliances using blueprints.

You've finished Module 1

Congratulations on completing Module 1.

If you are looking for additional information on Blueprinting Applications with vRealize Automation, try:

- Click on this link
- Or use your smart device to scan the QRC Code.

Proceed to any module below which interests you most.

- **Module 2 - Blueprinting and Managing Network Topologies with vRealize Automation** (60 minutes) (Advanced) Incorporate NSX network constructs to deliver end to end automation of your applications. From simple load balancing, to more complex topologies - consuming existing networks or deploying them at runtime. You will also explore some of the Day 2 capabilities to manage your NSX constructs.

- **Module 3 - Using vRealize Automation to Discover, Deploy and Manage Containers** (60 minutes) (Advanced) Whether you want to provide container hosts in controlled environments, or discover and manage containers that already exist within your environment, we have you covered. This module will give you everything that you need to get started with Container Management through vRealize Automation.

- **Module 4 - Working with the vRealize Automation Health Broker** (45 minutes) (Advanced) Monitoring your environment is a key part of system
administration. This module will show you how to take advantage of the new Health Broker service to monitor key parts of your vRealize Automation environment, and highlight any issues that would result in degradation of service. You'll even get to build and run your own custom monitoring solutions!

**How to End Lab**

To end your lab click on the **END** button.
Module 2 - Blueprinting and Managing Network Topologies with vRealize Automation (60 minutes)
Introduction

In this lab you will practice creating advanced blueprints that incorporate Network and Security components.

This lab contains the following lessons:

• Introduction
• Getting Prepared
• Working with NSX Constructs in vRealize Automation
• Day Two NSX Operations
• Conclusion

If you need additional time to complete this lab, you may click the Extend button at the top of the Hands On Labs interface.
Getting Prepared

This lesson is a quick one - it just validates that you have a working Blueprint to begin introducing networking and security constructs to your application.

Open Chrome Browser from Windows Quick Launch Task Bar

1. Click on the Chrome Icon on the Windows Quick Launch Task Bar.

Login to vRealize Automation

1. Enter cloudadmin as the username and VMware1! as the password.
2. Click Sign in.
Check Your Blueprint Inventory

1. Click on the Design tab.
2. Select the Blueprints menu.

If you completed Module 1, then you should see the Two Tier Wordpress Blueprint as shown above. If you were able to successfully create and request that blueprint, please skip ahead to Open Your Blueprint to carry on with the lab.

If you created the Blueprint, but it didn't provision correctly, or you started this lab at this module and don't see the Two Tier Wordpress Blueprint, then please click through to the next step to import a working copy of the Blueprint.

Launch CloudClient

1. Minimize the web browser, and launch CloudClient from the desktop. It will take a moment to load completely.

It is important to note that when using CloudClient, roles are honored in the same way as they would be in the UI. This means that if you want to import content, you need to have the right role. This will differ for XaaS Blueprints (XaaS Architect), IaaS Blueprints (Infrastructure Architect) and Software Components (Software Architect/Infrastructure Architect).
Login to vRealize Automation

One of the big challenges with working on command line based tools is using them regularly enough to remember the command sequences! Thankfully CloudClient supports tab based autocomplete.

1. Type in **vra login userpass --** and then press tab. You’ll note that a full list of arguments is provided.
2. Complete the command to be as shown below and then press Enter.

   vra login userpass --server vra-01a.corp.local --user cloudadmin@corp.local --password VMware1! --tenant vsphere.local

---

HOL-1921-02-CMP
Import Blueprint

When performing an import, you have a few options. How do you want to handle conflict resolution - skip or overwrite? Would you like to perform a dry run and not actually import anything, just confirm that the import will work ok?

If you'd like to take a look at the available arguments, then you can use the tab autocompletion. In this case, let's keep it simple:

1. Type the following command and press enter.

```
vra content import --path c:\hol\hol-1921-02-cmp\TwoTierWordpress-composite-blueprint.zip --resolution OVERWRITE --precheck WARN
```

You should see the same response under Import Result (SUCCESS) if it has imported successfully.

2. Close the CloudClient by typing exit and hitting Enter

Check for Imported Blueprint
1. Navigate to the **Blueprints** page by clicking on the **Design** tab, and then **Blueprints**.

2. If your import worked correctly, you should be able to see a **Two Tier Wordpress** Blueprint.

With that, you are ready to begin adding network and security components to your Blueprint.
Working with NSX Constructs in vRealize Automation

The integration between vRealize Automation and NSX can be utilized in a number of ways. The most simple is to deploy virtual machines onto pre-provisioned Logical Switches, and treat them in the same way as we would portgroups - as an External Network.

Another option is to make the network lifecycle part of the application lifecycle, and introduce networks and load balancers that are deployed at request time, and torn down when the application is. Let's take a closer look at the integration with NSX.

Open Your Blueprint

1. Select the hyperlink of your Two Tier Wordpress Blueprint to open the design canvas.
Since you are working with limited resolution, be sure to take advantage of the zoom button in the top right hand corner of the screen. You can use the scroll bars to navigate around while zoomed in, but if you want to review your topology then the zoom out button will be your best friend.
Add a Security Group to the Canvas

1. Select the **Network & Security** category.
2. Drag an **Existing Security Group** onto the canvas.
3. Click on the **Security group** ellipses to map the tag object to an NSX Security Tag.

Assign the Security Group

1. Select the **PROVIDER-HTTP** Security Tag.
2. Click **OK**.
3. Click somewhere on the Design Canvas to close the configuration dialog (not shown).

### Add another Security Group to the Canvas

1. Drag an **Existing Security Group** onto the canvas.
2. Click on the **Security group** ellipses to map the object to an NSX Security Group.

### Assign the Security Group

1. Choose the **PROVIDER-MYSQL** Security Group.
2. Click **OK**.
3. Click somewhere on the Design Canvas to close the configuration dialog (not shown).
Configuring Web Server Security

Now that the security objects are on the canvas, you need to map them to the appropriate virtual machine components just like you did with the network.

1. Select your Web_Server virtual machine
2. Click the Security tab.
3. Place a tick in the PROVIDER-HTTP checkbox.

Click on the canvas to exit the configuration page.

Configure Database Server Security

1. Select the Database_Server
2. Click the Security tab
3. Place a tick in the PROVIDER-MYSQL check box.
Click on the canvas to exit the configuration page.

**A Matter of Style**

![Design Canvas](image)

Depending on where you placed your Security Tag and Security Group, you may have an easy to decipher Blueprint, or one that is starting to look like the picture above.

Changes to the canvas layout are best performed when the canvas zoom is at 100%. It is up to you whether you make changes at this point, but once you start to build more complex Blueprints in your own environment it is definitely worth considering where you place objects in order to make them easily decipherable for others. The picture below shows a cleaner layout.
Add an On Demand NAT Network to the Canvas

1. Select "On-Demand NAT Network" from the Categories panel.
2. Create a new On-Demand NAT Network profile.

Description:

* ID: On-Demand_NAT_Network_1
* Parent network profile: 

[Diagram showing the design canvas with selected On-Demand NAT Network]
1. The next step is to add some dynamically created networking to your Blueprint. Drag an **On-Demand NAT Network** onto the canvas.
2. Click the **Parent network profile** ellipses to map the network across.

### Assign the On Demand NAT Network

1. Select the **One-to-Many NAT** profile.
2. Click **OK**.
3. Click somewhere on the Design Canvas to close the configuration dialog (not shown).
Configure Virtual Machine Networking

Like the security components in the previous steps, you need to assign the network construct to the virtual machine components.

1. Click on the Web_Server component, and then select the Network tab in the configuration dialog.
2. Double click on the network adapter to make it editable, and from the Network dropdown, select OnetoManyNAT.
3. Click OK.
4. Repeat this process for the Database_Server.
5. Click on the Design Canvas to close the configuration dialog (not shown).

The Network dropdown is populated with all of the available networks that have been added to the Blueprint through the Design Canvas.

Add an On Demand Load Balancer
The final element required for your application is a Load Balancer. While you are only using a single node for both the Web and Database tiers, validating the load balancer function before you scale is a good approach.

1. Drag an **On-Demand Load Balancer** onto the canvas.

**Configure General Settings**

1. From the **Member** dropdown, choose **Web_Server**.
2. From the **Member network** dropdown, choose **OnetoManyNAT (NIC 0)**.
3. From the **VIP Network** dropdown, choose **DefaultExternalNetworkProfile**.
4. Click on the **New** button to create a new **Virtual Server** on the load balancer.

Note: The OnetoManyNat Network profile selected earlier has its external network connected to the DefaultExternalNetworkProfile. These two must match to allow the Edge to serve the network properly.
Configure Virtual Server

1. Click on **Customize** to edit the settings.

Adjust the Health Check Settings

1. Select the **Health Check** tab
2. Set the **Interval** to 30 seconds
3. Set the **Timeout** to 45 seconds
4. Set the **Max Retries** to 5
5. Click **OK**
6. Click on the Design Canvas to close the configuration dialog (not shown).

**Update Site IP Mapping**

The final step is to make sure that Wordpress is going to use the IP of the Virtual Server (VIP) for the website and any links. Otherwise it would attempt to send all client traffic to its own address, which is not exposed since it is being NATed.

1. Select the **Wordpress_Siteconfig** component on the **Web_Server**, then select the **Properties** tab.
2. Choose the **site_ip** attribute, click **Edit** and place a tick in the **Binding** checkbox if it isn't already there.
3. In the **Value** field, edit the value to be **On-Demand_Load_Balancer_1-vip_address**
4. Click **OK**
5. Click **Save** and then click anywhere on the canvas to exit the configuration dialog.
Review and Save

Take a moment to bask in the glory of creating a complete application topology.

1. Once you finish basking, click **Finish**.
Entitle Your Blueprint

1. Click on the **Administration** menu.
2. Select **Catalog Management** (Not shown), and then **Catalog Items**.
3. Locate your **Two Tier Wordpress** blueprint in the list and click on the hyperlink to open up the Catalog Item.
Add Catalog Item to Service

1. Using the Service drop down, assign the Catalog Item to the Infrastructure Service.
2. Click on OK to save your changes and exit.
   Finally, you can request your new shiny Wordpress Application
3. Logout from your cloudadmin session using the Logout button in the top right corner, then click the Go back to login page button (Not shown)

Login as DevUser

1. devuser
2. Corp.local

Sign in
Forgot password?
Change to a different domain
vmware
1. Enter **devuser** as your username, and **VMware1!** as your password.
2. Click the **Sign in** button.

**Request Blueprint**

1. Navigate to the **Catalog** tab.
2. Locate the **Two Tier Wordpress** Catalog Item and click on **Request**.
Set Site Name

1. Click on the **Web_Server**.
2. Set the **SiteTitle** value to **Succulent**.
3. Click **Submit**.

Monitor Your Request Progress

1. Keep an eye on the progress of your request by navigating to the **Requests** tab.
2. Watch the Status value, until it changes to Successful. (Note: click the refresh icon at the bottom of the screen to refresh the status. This could take 10-15 minutes)
3. Click on the Items tab.

Get IP Address

1. Expand out your Two Tier Wordpress Deployment and identify the IP Address of your On-Demand_LoadBalancer.

Validate Provisioning
1. Enter the IP Address into a new browser window and ensure that you can access the site.
Day Two NSX Operations

Introduction

Well, your request has provisioned - but what happens when you decide to do something crazy like apply a certificate to your website and traffic changes from HTTP on port 80 to HTTPS on 443? Moving to https would be painful if you had to go and ask someone else to update your Load Balancer configuration. Similarly, adding or removing security groups that are associated with that change could also be time consuming if you had to get someone else to do it for you.

What if you could make these changes yourself? The great news is that you can do this yourself if you have been entitled to the appropriate Day 2 Actions.

Change Security

To explore the new Day 2 NSX capabilities, navigate to your deployment.

1. Click in the Items tab.
2. Select your Wordpress Deployment. Note that the random ID appended to the end of your deployment may not be identical to the screenshot.
3. Click on Actions and select Change Security.
Remove Security Group

1. Select the Web_Server.
2. Click on the assigned Security Group PROVIDER-HTTP.
3. Click on the Remove button.
4. Click Yes (not shown).

Add Security Group

1. With the Web_Server still selected, click on Add > Existing Security Group.
Choose Your Security Group

You will be presented with a list of the available Security Groups from NSX.

1. Place a mark in the PROVIDER-HTTPS checkbox.
2. Click OK, and then click Submit.
3. Click OK on the confirmation (not shown).

Navigate to vCenter

To validate that this change has worked successfully, you will need to take a look at the Security Groups in vCenter. Since the NSX plugin is not available in the HTML5 client, you will need to login to the Web Client.

1. Open a new tab, and click on the Region A bookmark link, then click on RegionA vCenter.
Login as Administrator

1. Place a checkmark in the **Use Windows session authentication** box.
2. Click **Login**.

Launch NSX Plugin

1. Click on the **Home** menu button.
2. Select the **Networking & Security** link to launch the NSX Plugin.
Review Security Groups

1. Select the Groups and Tags.
2. Click on Security Group and review the memberships of each group by selecting them and then clicking the Edit button.

You'll note that your virtual machine is added as "Static include member". This simply means that it has been inserted as a static object, rather than being collected dynamically as it would be if it was associated with a Security Tag and added to a Security Group as a Dynamic Inclusion. Static inclusions are far more efficient at a computational level, and in an environment of scale would be the recommended architectural approach to take.

Leave this tab open, and move back to vRealize Automation.

Updating Your Load balancer Configuration
Having allowed HTTPS traffic to flow to your Web Server, you still have a task to take care of - updating your load balancer to pass HTTPS traffic.

1. In your deployment, locate and select your load balancer. Remember that the objects may be in a different order than shown here.
2. Click on **Actions > Reconfigure**.

### Add Virtual Server

There are a few options - create a new Virtual Server, edit an existing Virtual Server, or delete a Virtual Server. In this case, you will add a new one. This will allow you to see both configurations in NSX Manager once you submit your request.

1. Click on **New**.
Configure General Details

1. Change the **Protocol** dropdown to **HTTPS**.
2. Under **Settings** click in the **Customize** radio button.
3. Click on the **Distribution** tab.

Configure Distribution Details

1. Ensure that the **Algorithm** is set to **ROUND_ROBIN**.
2. Set the **Persistence** to **Source IP**.
3. Set the **Expires in (sec)** value to 600.
4. Click the **Health Check** tab.

**Review Health Check Options**

![Health Check Options](image)

You won't actually change anything on this tab. It is worth looking at the various options you have to configure - as you can see, the exposed capabilities are equivalent to those that you would get access to via the NSX plugin in vCenter.

1. Click on the **Advanced** tab.
Review Advanced Options

Again, you won't make any changes here. Just review the options that you have available.

1. Click on OK to return to the previous screen

Submit Your Request

1. Click on Submit to send the request, then click OK
Select the Edges Menu

1. Choose **NSX Edges** to display all Edge devices under management.
2. **Double click** the Edge providing your load balancing. Look for the name "Edge-TwoTierWordpress" or similar to identify your dedicated Edge device.

Return to the vCenter tab

Review Virtual Servers

1. Select the **Load Balancer** tab.
2. Click on the **Virtual Servers** menu.

   1. Choose **NSX Edges** to display all Edge devices under management.
   2. **Double click** the Edge providing your load balancing. Look for the name "Edge-TwoTierWordpress" or similar to identify your dedicated Edge device.

   Return to the vCenter tab

Review Virtual Servers

1. Select the **Load Balancer** tab.
2. Click on the **Virtual Servers** menu.
3. Click on the second Virtual Server (for https).
4. Take note of the Application Profile value.

**Select Application Profile**

1. Select the Application Profiles menu.
2. Select the Application Profile identified in the previous step. This will be the one that has a type of HTTPS.
3. Click on the Edit button to open it.

**Validate Application Profile**
Take a look at the configuration of the Application Profile, and validate that it matches the configuration you set when updating the Load Balancer details in vRealize Automation.

1. Click Cancel (Not Shown)

**Destroying Your Deployment**

![Image of vRealize Automation interface]

Now that you've finished reviewing the NSX configuration, it is time to tidy up and move onto the next lesson. Head back to your vRealize Automation page.

1. Click on the **Items** tab.
2. Select your Deployment, click on **Actions** and then click **Destroy**.

On the page that pops up, click the **Submit** button (not shown) to confirm your request. Once that completes, click on the **Logout** button in the top right corner of the browser (also not shown).
Conclusion

Now that you have spent some time working with Networking and Security constructs, you should be comfortable with building out applications complete with all of the required connectivity to make them available to both the outside world, and each other.

You've finished Module 2

Congratulations on completing Module 2.

If you are looking for additional information on using NSX constructs within vRA blueprints, try one of these:

- Click on this [link](#)
- Or use your smart device to scan the QRC Code.

Proceed to any module below which interests you most.

- **Module 1 - Blueprinting Applications with vRealize Automation** (60 minutes) (Advanced) Discover what lies beyond IaaS, and how you can use blueprint parameterization and Software Components to deliver real world benefits over and above simple virtual machine provisioning. Learn how to handle dependencies and pass information between components in multi-tier applications, and parameterize more complex scripts to allow for reuse across blueprints.
- **Module 3 - Using vRealize Automation to Discover, Deploy and Manage Containers** (60 minutes) (Advanced) Whether you want to provide container hosts in controlled environments, or discover and manage containers that already exist within your environment, we have you covered. This module will give you everything that you need to get started with Container Management through vRealize Automation.
- **Module 4 - Working with the vRealize Automation Health Broker** (45 minutes) (Advanced) Monitoring your environment is a key part of system administration. This module will show you how to take advantage of the new Health Broker
service to monitor key parts of your vRealize Automation environment, and highlight any issues that would result in degradation of service. You'll even get to build and run your own custom monitoring solutions!

**How to End Lab**

To end your lab click on the **END** button.
Module 3 - Using vRealize Automation to Discover, Deploy and Manage Containers (60 minutes)
Introduction

This Module contains the following lessons:

- Building Your First Container Based Application
- Sharing Your Application
- Conclusion
Building Your First Container Based Application

In this section you will learn about the basics of creating a container based application, including subscribing to a private registry.

Open Chrome Browser from Windows Quick Launch Task Bar

1. Click on the Chrome Icon on the Windows Quick Launch Task Bar.

Login to vRealize Automation

1. Enter cloudadmin as the username and VMware1! as the password.
2. Click Sign in.
Set Browser Zoom

The Admiral UI is built for a modern screen, and doesn't display well at the default resolution.

1. Select the Customization button.
2. Set the zoom to 75%.

Adjust this value as necessary through the lesson if you want more screen real estate, or alternatively more detail.
Let's start by adding a container host

1. Select the **Containers** tab
2. Select **Clusters**
3. Click **+New**
Enter Container Host Details

1. In the Name field, enter the following: vch-01a.corp.local
2. Select **Docker** from the drop down list, to specify this is a docker host
3. In the Address field, enter the following: http://vch-01a.corp.local:2375
4. Click **Save**
You may notice that the state of the container host is showing as unknown. If that is the case,

1. Select the 3 dots to open the menu and select Rescan. This process is extremely quick to complete.
2. When it completes, the Containers value will change to 7.

Let's take a closer look at the container host to find out what they are.
Access Your Container Host

Before we take a look at these containers in Admiral, let’s do it the old fashioned way.

1. Launch Putty using the shortcut on the taskbar (not shown).
2. Scroll down until you see `vch-01a.corp.local` and select it.
3. Click **Load**.
4. Click **Open**.

Check for Containers

Resize your Putty session so that it is the full width of your screen.
1. Type in `docker ps` and hit enter.
2. Review the containers, and pay particular attention to the **Status** column.

Close the putty session and head back to the vRealize Automation interface.

## Adding a Container Registry

![Image of vRealize Automation interface showing how to add a container registry](image)

A container registry is useful for a number reasons. The Hands On Labs actually simulates a production environment quite well, in that there is no Internet access from the vRealize Automation Appliance. In that case, how can you go about getting container images? The answer is to host a registry internally.

Another benefit to hosting your own registry is that you can provide your own hardened and tested images for consumption.

VMware provides an open source registry called Harbor, which is what you will setup a connection to;

1. Select **Registries** on the menu
2. Click on the **+Registry** button
Configure Registry Details

Enter the Registry details;

1. Enter the following details:
   Address: **https://vch-01a.corp.local:443**
   Name: Harbor
   Login Credential: Harbor (selected from the dropdown).
2. To validate the connection, click **Verify**, you should see a green bar stating **Verified successfully**
3. If verification succeeds, click on **Save**
Check Your Registry for Images

1. Open a new tab in your browser, and navigate to [https://vch-01a.corp.local](https://vch-01a.corp.local) using the bookmarked link in HOL Admin.
2. Enter **admin** as your username, and **VMware1!** as your password.
3. Click the **Log In** button.
Review Projects

While this lab doesn't provide a deep dive on the capabilities of Harbor, it is worth noting that you can create individual projects, each with their own set of users, roles and repositories. Check out Harbor on Github for more information.

1. Click on the hol Project.
Review Repositories

The hol Project contains two Repositories. One for MySQL and one for Wordpress.

1. Click into the hol/mysql Repository to see its contents.

Review MySQL Repository Contents
This repository only contains a single mysql image, but it could contain a number of different versions if required.

You'll also see the pull command that can be used if you want to work with Harbor directly through Docker. It is important to note that you do not have to use vRealize Automation or Admiral to get value out of Harbor as a registry.

Head back over to the vRealize Automation tab, but leave this one open. You'll be coming back before too long.

**Building an Application**

Now that you know what your Registry contains, it's not going to come as much of a surprise what the application is that you're going to build!

1. Select **Applications** from the menu
2. Click on **+Application**.
Specify Application Name

1. Give your application a name - type in Wordpress.
2. Click on Proceed.

Add Your First Container

1. Hover over the grey plus symbol on the new tile until it expands out to the three available components. Click on Add Container.
Find MySQL

1. In the search dialog, type in `mysql` and hit enter.
2. When the image appears, click **Continue**.

Set MySQL Environment Values

1. In the search dialog, type in `mysql` and hit enter.
2. When the image appears, click **Continue**.

Set MySQL Environment Values

1. In the search dialog, type in `mysql` and hit enter.
2. When the image appears, click **Continue**.
1. Select the **Environment** tab where you can define startup **environment variables**.
2. Add **MYSQL_ROOT_PASSWORD** and **VMware1!** as the **Name** and **Value** respectively.
3. Click **Add**

**Adding a Second Container**

1. Hover over the grey plus symbol on the new tile until it expands out to the three available components. Click on **Add Container**.
Search for Wordpress

1. In the search dialog, type **wordpress** and hit enter.
2. When the image appears, click **Continue**.

Set Wordpress Cluster Size

1. In the search dialog, type **wordpress** and hit enter.
2. When the image appears, click **Continue**.
1. Click on the Policy tab.
2. Update the Cluster Size value to 2.
3. Scroll to the bottom of the page and click Add (Not shown)

**Add a Dependency Link**

You can add a dependency link between two (or more) containers very simply and easily.

1. Hover your mouse over the Wordpress image, and click on the plus symbol to create a link. You will configure this in the next step.
Configure the Link

1. From the **Add a link** dropdown, select **mysql**.
2. Click on the **Tick** to save the link.

Choosing an Option

Once you've finished assembling your application, you have four different things that you can choose to do with it. The buttons for each option are in the top right corner of the interface.

From left to right, they are:

- Provision
- Publish
- Export
- Delete

1. For now, go ahead and click the **Provision** button. You will have a chance to both Publish and Export in a later lesson.
Taking a Shortcut

On the right hand side of the screen, you will notice that a Requests tab has opened to track the task. The provisioning process will take a few moments to complete.

1. Once your request completes successfully, you can click on the eye symbol and it will take you straight to the object associated with the request. In this case it is your application. Go ahead and click it.
What Now?

Having followed the link from in the previous step, you can now see your application, and if you look carefully you will see the search criteria that was used to find it.

Note: The guid for both the search and the application will not be identical to that shown in the image above.

1. Click into the application to view the details.
A Little Investigation

Once you are inside the application, you can see the two different tiers.

1. Click into the **Scaled Containers**. This is a construct created by Admiral to represent a cluster of containers. Remember configuring the cluster size for the wordpress image to be two? Inside this object you will find two instances of the Wordpress container.
2. Once inside Scaled Containers, click on one of the Wordpress Child Containers (not shown)
Searching for Proof

Once you get through the Scaled Container object and down to the container level, you can see a fair bit of information about what is going on. You can view resource consumption, container metadata, and even logs!

1. Scroll down the properties and click on the **Ports** URL. This will open the URL and allow you to see the published application.
Result!

Look at that - it works!

In the next lesson, you will have the opportunity to publish your application for consumption through the vRealize Automation catalog. You will also be able to export it out to docker-compose compatible YAML.

1. Close this browser tab and return to the vRealize Automation browser tab to continue.
Sharing Your Application

In this lesson you will learn about the different ways you can share your application after it has been built.

Whether that is exporting it as YAML for consumption with docker-compose, publishing to vRealize Automation so it can be requested through the central catalog, or exported as vRealize Automation compliant YAML.

To do that, you are going to need to find the application you built in the previous lesson.

Find Your Application

1. Select the **Templates** tab on the left hand side.
2. Next, click on **Wordpress**
Remember These?

Last time you looked at these icons, it was to provision your application. This time, you are going to export it as YAML.

1. Click on the **Export** button.

Export Your Application

1. Select **YAML Blueprint**.
2. Click **OK**.

Repeat the process for the **Docker Compose** export.

Open the YAML Files

1. Open Windows Explorer and navigate to your **Downloads** folder.
2. Highlight both of your YAML files, right click and select **Edit with Notepad++**. They will open in the same editor, in separate tabs.

**Compare the Pair**

Have a look at both of the YAML files. You'll notice that the docker-compose YAML has fewer lines of code. If you look closely at the Blueprint YAML, you will see that it contains some additional information. The **type** is important to vRealize Automation as it works with many different component types, from AWS machines, to NSX components, through to - as in this case - containers. The file also includes the **Deployment Policy ID**. Another piece of metadata that is specific to vRealize Automation, in order to provide a level of control over the placement of the application components.

Both formats have their place, and it's important to understand for whom and for what purpose you are exporting to make sure that you choose the right format.

Close **Notepad++** and return to vRealize Automation to continue working.

**Publishing Your Application**
Getting your application into the vRealize Automation blueprint library is a very straightforward process. No export/import required, just the click of a button.

1. Click on the **Push** button. Once it completes you will see **Pushed Successfully** appear at the top of the screen.

**Navigate to the Blueprints Tab**

1. Select the **Design** tab.
2. Click on the **Blueprints** menu.
3. Select your **Wordpress** Blueprint, but don't click on the hyperlink.
4. Click on the **Publish** button.
5. Finally, take a look at how your application looks on the design canvas by clicking **Edit**.

**Review Your Application on the Design Canvas**

You'll note that this looks similar to a virtual machine based Blueprint. The **Wordpress** component shows as a cluster, and the dependency mapping is in place to show that **Wordpress** can't be deployed until **MySQL** is online.
Click into the **wordpress** component and review the **General** and **Policy** tabs. Notice that items such as **Image** and **Deployment Policy** are configured as you defined them in the Containers tab.

As you can see, the full set of metadata has come along with your application. Let's carry on and publish it to the catalog.

1. Click **Cancel** at the bottom of the Design Canvas to move on (not shown).

**Entitle Your Application**

1. Select the **Administration** tab.
2. Click on the **Catalog Management** menu (not shown)
3. Click **Catalog Items**.
4. Select the **Wordpress** Catalog Item hyperlink.
Assign to a Service

1. Scroll to the bottom of the page, click on the Service dropdown and choose Infrastructure.
2. Click OK to save your change.

The next step is to request the Catalog Item. To do that you will need to login as a different user. Click on the Logout link in the top right corner of your screen, followed by the Go back to login page button (not shown).

Login as DevUser

1. Enter devuser as your username, and VMware1! as your password.
2. Click the **Sign in** button.

**Request Catalog Item**

1. Select the **Catalog** tab.
2. Locate your **Wordpress** Catalog Item and click **Request**.
Submit Request

1. Leave the default values for all fields and click **Submit**.
2. Click **OK** on the Request submission (not shown).

Monitor Your Request

1. Click on the **Requests** tab to monitor the status of your request. It should take no more than two minutes to complete.
2. When the **Status** changes to successful, click on the **Items** tab.

Note: You may need to click the **Refresh** button in order to update the **Status** (not shown).
Inspect Your Deployment

1. Click the triangle to expand your deployment.
2. Locate one of the two Wordpress components, and click its hyperlink.

Review Container Detail
Does this look familiar? This is the exact same UI as you had when inspecting a container in the Containers tab. All the same information is there, from resource consumption to logging.

1. If you are so inclined, click on the **Ports** link to confirm that Wordpress has been successfully deployed.
2. Click **Close** when you are finished

### Destroy Your Deployment

To free up resources for the next lesson destroy your Wordpress application.

1. Click to the right of the wordpress deployment name to select this deployment.
2. Select **Destroy** from the Actions menu.
3. Click **Submit** on the resulting page (Not shown)
4. Click **OK** (Not shown)

Finally, in the top right corner of the screen, click **Logout**.

### Resetting Your Zoom
Don't forget to reset your zoom back to 100%.

1. Select the Customization button.
2. Set the zoom to 75%.
Conclusion

Now that you've gained a basic understanding of how to work with containers in vRealize Automation, you should be able to create and deploy new applications using images from private or public registries.

You've finished Module 3

Congratulations on completing Module 3.

If you are looking for additional information on using containers with vRealize Automation, try one of these:

- Click on this link
- Or use your smart device to scan the QRC Code.

Proceed to any module below which interests you most.

- **Module 1 - Blueprinting Applications with vRealize Automation** (60 minutes) (Advanced) Discover what lies beyond IaaS, and how you can use blueprint parameterization and Software Components to deliver real world benefits over and above simple virtual machine provisioning. Learn how to handle dependencies and pass information between components in multi-tier applications, and parameterize more complex scripts to allow for reuse across blueprints.

- **Module 2 - Blueprinting and Managing Network Topologies with vRealize Automation** (60 minutes) (Advanced) Incorporate NSX network constructs to deliver end to end automation of your applications. From simple load balancing, to more complex topologies - consuming existing networks or deploying them at runtime. You will also explore some of the Day 2 capabilities to manage your NSX constructs.
Module 4 - Working with the vRealize Automation Health Broker (45 minutes)  
(Advanced) Monitoring your environment is a key part of system administration. This module will show you how to take advantage of the new Health Broker service to monitor key parts of your vRealize Automation environment, and highlight any issues that would result in degradation of service. You'll even get to build and run your own custom monitoring solutions!

How to End Lab

To end your lab click on the END button.
Module 4 - Working with the vRealize Automation Health Broker (45 minutes)
Introduction

In this module you will learn how to use the Health Service to create tests to manage the health of your vRealize Automation environment.

This module contains the following lessons:

- Lesson 1 - Creating a System Test
- Lesson 2 - Creating a Tenant Test
- Lesson 3 - Validating Tenant Health
- Conclusion

If you need additional time to complete this lab, you may click the Extend button at the top of the Hands On Labs interface.
Review the System Test

The starting point when managing the health of your vRealize Automation environment is keeping an eye on those things that would affect not just a single tenant, but all of them. This is where System Tests come into play.

In the Hands on Lab environment we have already set these up for you. Although the tests are quick and simple to set up, we needed to create them in advance to allow us to use them as a means to checking that the Lab environment is healthy and in a ready state to ensure an excellent experience for yourselves.

Open Chrome Browser from Windows Quick Launch Task Bar

The easiest way to get started with the Health Service is through the vRealize Automation Web UI. Let's open up a browser and login to vRealize Automation.

1. Click on the Chrome Icon on the Windows Quick Launch Task Bar.

Login to vRealize Automation

1. Login as cloudadmin, with the password VMware1!
2. Click the Sign in button.
A lot of the screens you are about to use won't display well at low resolutions. To work around that, adjust your browser zoom to 75%.

1. Click the ellipses in the top right hand corner of the browser.
2. Click the "-" symbol until the zoom value changes to 75%.
View the Health Tests

1. Navigate to the Administration menu.
2. Scroll to the bottom of the menu and select Health.

On this screen you will see 3 tiles representing 3 different sets of tests. These tests are all available out of the box and are included with vRA

3. On the vRA System Test click on Edit so we can view the configuration
View Basic Configuration

1. On this screen you can see some basic information about the test - a name, description, the product to run the test against, and the frequency at which the test should run.
2. Click on the Next button.

View the selected Test Suites
There are two suites of tests available to run against vRealize Automation - System Tests, and Tenant Tests.

System Tests check for underlying infrastructure related issues, such as certificate mismatches, storage consumption, and connectivity between the nodes.

Tenant Tests (which we will look at shortly) look for issues within the system that would affect a given tenant.

1. Click **Next**.

**Review the Configuration Parameters**

![Configuration Parameters](image)

In order to perform the system tests, some details about vRealize Automation and accounts with the appropriate permission levels are required.

1. Review this information
2. Click **Cancel**
Running the Test

Now that you have reviewed the System Test settings let's go ahead and run the test.

1. To kickoff the initial test run, click on the **RUN** button.
2. Wait for the progress bar to complete its march to the right of the tile.

Review the Results

The tile should update to contain a **Green Circle** pictogram.

1. Click on the tile so that we can take a look at the test details.
Have a look through the different checks to familiarise yourself with what is covered by the System Test. As you can see, all of the checks are related to items that would impact the entire system.

1. When you have finished looking through these, click < Back.
Review the Tenant Test

Having reviewed a test that looks for system wide issues, it's now time to take a more focused look at the health of a given tenant.

View the Tenant Test

1. On the vSphere.Local Tenant Test tile click **Edit**

View the Basic Configuration
1. On this screen some basic information about the test is provided— a name, description, the product to run the test against, and the frequency at which the test should run.
2. Click on the Next button.

**View the selected Test Suites**

1. This time the Tenant Tests Suite is selected.

Tenant Tests look for issues within the system that would affect a given tenant.

As a refresher, System Tests check for underlying infrastructure related issues, such as certificate mismatches, storage consumption, and connectivity between the nodes. You already viewed a System Test in earlier steps.

2. Click Next.
Review Configuration Parameters

In order to perform the tenant tests, we need to provide some details about vRealize Automation and accounts with the appropriate permission levels. You may notice that the Tenant tests require more configuration in comparison to System tests. This is due to additional account information specific to the tenant being required.

1. Review the parameters
2. Click Cancel

Running the Test
1. To kickoff the initial Tenant Test run, click on the Run button.

Wait for the progress bar to complete its march to the right of the tile.

**Review the Results**

As before, when it is complete you should see a Green Circle pictogram.

1. Click on the tile so that we can take a look at the test details.
Investigate the Checks

Have a look through the different checks to familiarise yourself with what is covered by the Tenant Test. As you can see, all of the checks are related to items that would impact the health of a specific tenant.

You may see some tests that have failed. This is due to the nature of the Hands on Lab environment, do not be alarmed.

1. When you have finished looking through the tests, click < Back.

Next up, we will go and introduce a problem in your environment and then run the test again.
Validating Tenant Health

To see some more useful feedback from the tests, you will need to introduce a failure within the environment. A simple example of this is to break connectivity to your vCenter endpoint.

Navigate to the Endpoints Screen

We are going to change the password of your vCenter Endpoint and see how the test responds.

1. Select the **Infrastructure** tab.
2. Click on **Endpoints**, then **Endpoints** again.
3. Click on the **vCenter** Endpoint to edit.
Change the vCenter Endpoint Password

1. Edit the password to something other than VMware1!
2. Click OK

You will see a notification that the Test Connection failed. This is expected since we just changed the password.
Run the Tenant Test Again

1. Select the Administration tab
2. Select the Health menu item on the left
3. Click Run to start the Tenant Test and wait for it to complete

When the test completes, the Green Circle should update, and you'll see a higher number of failed tests.

4. Click on the tile to open the test run details.
Review the Failed Checks

Note in this test that multiple items failed due to the password change in the vSphere endpoint.

1. Select **Failed** from the drop down list to filter on the failed results
2. Take time to review the failed items and then click < Back

Navigate back to the Endpoints Screen
Let's change the password of your vCenter Endpoint Back to the working value.

1. Select the **Infrastructure** tab.
2. Click on **Endpoints**, then **Endpoints** again.
3. Choose the **vCenter** Endpoint to edit.

### Change the vCenter Endpoint Password

1. Edit the password to be VMware1!
2. Click **OK**
Health Broker Extensibility

So now that you have had the opportunity to review the built in tests, let's take the opportunity to learn how to use the health broker extensibility to import your own tests.

In this lesson you will learn how to import a custom wheel and run your own tests.

Open the Health Broker Extensibility Page

The Health Broker service allows you to create your own tests and import them as a wheel. For example, wouldn't it be great to ensure the health of external services which vRA relies on. In this lesson you will be importing a test which will check connectivity to NSX. This will ensure that we are aware of any issues with NSX connectivity to understand if there will be a problem deploying blueprints containing NSX network constructs.

1. In the top right hand corner Click on the cog
2. Select Extensibility from the dropdown

Import the Custom Wheel

1. Click New Asset
Add Asset Information

Here you will add information to describe your new custom wheel

1. Enter the following information in the fields
   ◦ Asset Title: **Basic NSX Test**
   ◦ Asset Description: **Test NSX Connectivity**
   ◦ Asset Version: **1.0**

2. Click **Choose File**
Import the Custom Test Wheel

For this lesson we have created and compiled the custom test for you. The tests are created using python 3 and packaged using a wheel binary package format. You can find more detail on how to create your own custom tests here; vRA 7.4 Health Service Extensibility.

1. Navigate to C:\hol\HOL-1921-02-CMP\ and select basic_vrealize_nsx_test-0.0.1.dev0-py3-none-any.whl
2. Click Open

Check Installation of the Wheel
1. Check the Status of your imported wheel
2. Click Refresh until the Status changes to **Installed**
3. Click `< Back

**Configure the Custom NSX Test**

Now it's time to configure the Custom Test that you just imported

1. Click the **+New Configuration** button

**Provide Basic Configuration Information**

Enter the basic configuration information;

1. Type **NSX System Test** for the name
2. Select **VMware NSX 6.2.4** from the Product dropdown list. Note this selection is not available out of the box, this was something that was created within the custom wheel using Python
3. Select **None** from the dropdown list for Schedule, you will run the test manually later
4. Click **Next**
Select the Test Suite

1. Select the **NSX Tests** checkbox
2. Click **Next**

Configure Test Parameters

1. Enter the following details, please note that the fields may appear in a different order to those shown;
   - NSX Manager Address: `nsxmgr-01a.corp.local`
   - NSX Manager Username: `admin`
NSX Manager Password: **VMware1!**

2. Click **Next**

**Review the Summary**

1. Review the summary details and Click **Finish**

**Run the NSX Test**

1. Click **Run** and wait for the test to complete
View the Test Results

1. Click on the tile to view the test results

Review the Test Results

1. You will see that the custom wheel that you imported runs 2 tests;
   - Checks for NSX connectivity - This test takes the account details you entered and uses them to connect to the NSX REST API returning details about the NSX manager. Notice that this test is marked Critical, this is defined with the Python script
   - Check Username - this is a simple test to check that the username entered is 'admin'

2. Click < Back
Edit the Test Configuration

Okay, so let's update the test configuration to add to some incorrect information so that we can force the test to fail and examine the results.

1. Click **Edit** on the NSX System Test tile

**Edit the Test Parameters**

1. Click Next twice (not shown) until you get to **3. Configure Parameters** screen
2. Change the NSX Manager Username to **administrator**
3. Click **Next** then **Finish** (not shown) to accept the changes
Re-Run the NSX Test

1. Click Run and wait for the test to complete

The test will fail as we now have incorrect account details

View the Test Results

1. Click on the tile to view the test results
Review the Test Results

You will see that the tests have now failed, this shouldn't be a surprise as we intended it to be this way. Now that the tests have failed you will see links in the Remediation and Cause columns

1. Click the **Cause** link in the Checks NSX Connectivity row

Review the Cause

Here you can see the connection request that was sent along with a 403 response, meaning that access was denied.

This message is constructed within the Python code and therefore its important to provide useful information in your output to make it easier to troubleshoot the problem.

The error message here is expected behaviour as we provided fake account details

1. Click **OK**
Review the Test Results

This time let's take a look at the Check Username test

1. Click the **Cause** link in the Check Username row

Review the Cause

And there we have it, the test checked the fake account you provided earlier against what was expected.

1. Click **OK**

Logout
1. Sadly this module must come to an end, feel free to investigate the tests further but when you are ready click Logout and proceed to the module conclusion.
Conclusion

Now that you have been introduced to the Health Service, you should be able to create Tenant and System centric tests to stay ahead of the curve and address any issues in your own environment before they reach a critical point. You have also seen how the Health Broker extensibility function allows you to create your own tests.

You've finished Module 4

Congratulations on completing Module 4.

If you are looking for additional information on the vRealize Automation Health Broker, try one of these:

- Click on this link
- Or use your smart device to scan the QRC Code.

Proceed to any module below which interests you most.

- **Module 1 - Blueprinting Applications with vRealize Automation** (60 minutes) (Intermediate) Discover what lies beyond IaaS, and how you can use blueprint parameterization and Software Components to deliver real world benefits over and above simple virtual machine provisioning. Learn how to handle dependencies and pass information between components in multi-tier applications, and parameterize more complex scripts to allow for reuse across blueprints.

- **Module 2 - Blueprinting and Managing Network Topologies with vRealize Automation** (60 minutes) (Intermediate) Incorporate NSX network constructs to deliver end to end automation of your applications. From simple load balancing, to more complex topologies - consuming existing networks or deploying them at runtime. You will also explore some of the Day 2 capabilities to manage your NSX constructs.
• **Module 3 - Using vRealize Automation to Discover, Deploy and Manage Containers** (60 minutes) (Intermediate) Whether you want to provide container hosts in controlled environments, or discover and manage containers that already exist within your environment, we've got you covered. This module will give you everything that you need to get started with Container Management through vRealize Automation.

**How to End Lab**

To end your lab click on the **END** button.
Conclusion

Thank you for participating in the VMware Hands-on Labs. Be sure to visit http://hol.vmware.com/ to continue your lab experience online.

Lab SKU: HOL-1921-02-CMP

Version: 20190914-212129